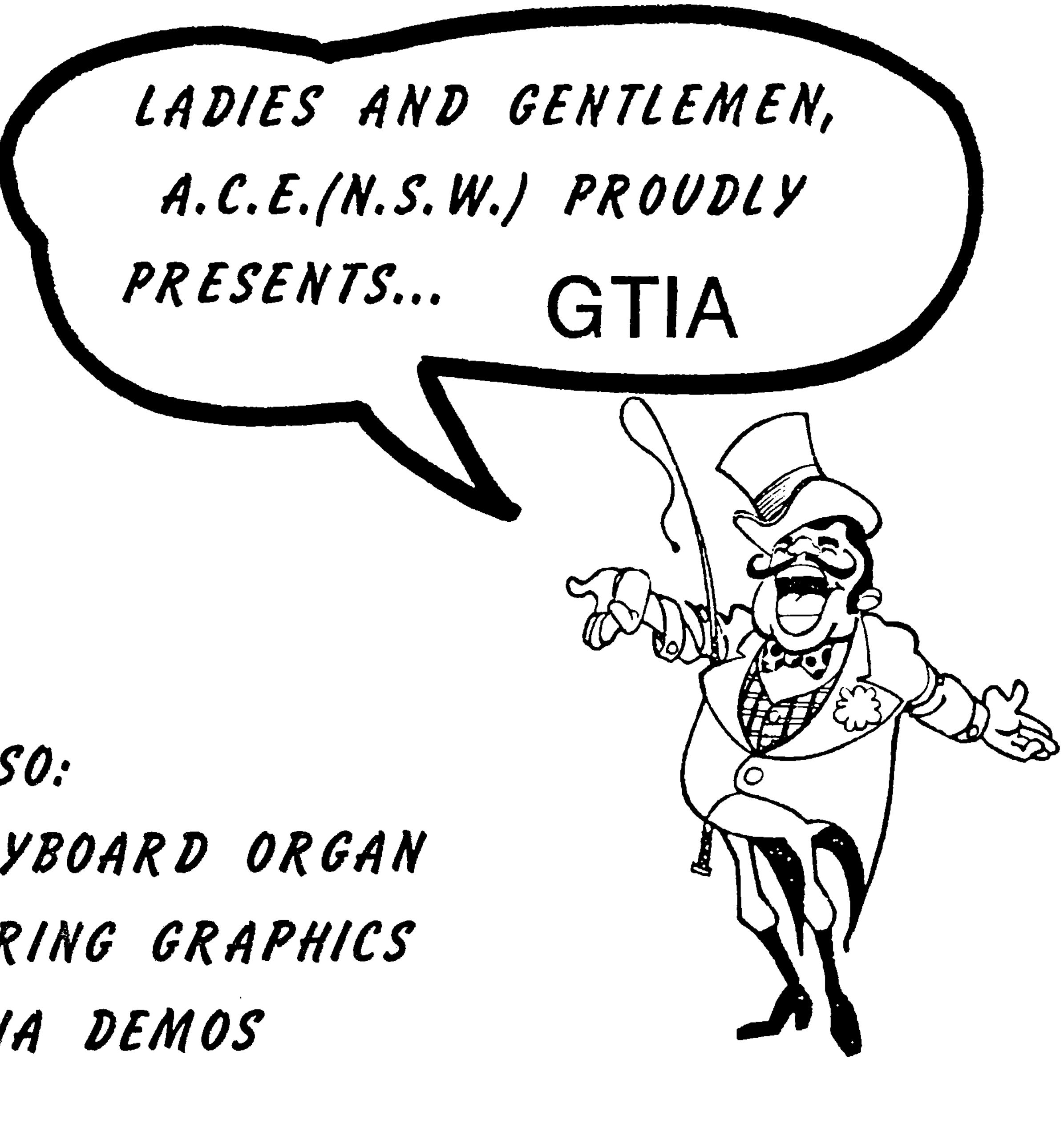
Atari Computer Enthusiasts (N.S.W.)

INSIDE INFO No. 3 October 1982


LADIES AND GENTLEMEN, A.C.E.(N.S.W.) PROUDLY PRESENTS... GTIA

ALSO:

KEYBOARD ORGAN

STRING GRAPHICS

GTIA DEMOS

Atari Computer Enthusiasts (N.S.W.)

Atari Computer Enthusiasts (N.S.W.) is an independent, non-profit computer users' group loosely affiliated with Atari Computer Enthusiasts in the U.S.A. We have no connections with ATARI, Inc. or their Australian distributors, Futuretronics Australia Pty Ltd.

Our aims include promotion of the ATARI 400/800 Home Computer System; instructing both beginners and advanced users in programming techniques; exchanging hints, tips and ideas amongst members and generally enjoying ourselves.

Meetings are held at 6.0 P.M. sharp on the first Monday of every month (or the second Monday if it clashes with a public holiday) at the offices of:

I.P. Sharp Associates
8th Floor,
Carlton Centre,
55 Elizabeth Street,
Sydney.
(between King Street and Martin Place)

Membership to A.C.E.(N.S.W.) is $15 joining fee and $15 annual subscription (or $10 joining fee and $10 annual subscription for students under 18 and still at school).

Subscriptions or postal enquiries may be directed to:

Atari Computer Enthusiasts (N.S.W.)
78 Ayres Road,
St. Ives,
N.S.W. 2075

Phone enquiries:

Garry Francis: 2-0933 Ext 354 (Work) or 789-1397 (Home)
Ron Baxter: 81-2388 (Home)
Paul Phillips: 449-6286 (Home)

☆☆☆

Meeting Dates for Remainder of 1982

11th October
1st November (A.G.M.)
6th December

New Members

Helen Abberfield, Cardiff North
David Anderson, Hornsby
Alan Cashin, Armidale
John Damen, Cromer
Nigel Delaforce, Everton Park, Qld.
Joe Delman, North Sydney
Peter Dickeson, Northbridge
Chris Fitzgerald, Clyde
Richard Gander, Pagewood
M. Gatica, Strathfield
Andrew Hallahan, Monterey
David Heap, Gymea
Andrew Jones, St. Ives
G. Koch, West Footscray, Vic.
Robert Lang, West Pymble
Alan Lee, Wickham, W.A.

Jeffrey Maddock, Hurlstone Park
J. Masters, Howick, New Zealand
Kerry McGill, Katoomba
Riccardo Modesto, Darling Point
Ian Murray, Waverley
Lewis Pallu, Bundaberg, Qld.
Wayne Randaud, Redhead
Richard Rannard, Northbridge
Ken Scalley, Winston Hills
Jilles Sheedy, Hardy's Bay
Kenneth Shiu, Lugarno
Steven Sommer, Mosman
Glen Stewart, Gorokan
Philip Strong, Braddon, A.C.T.
Ernie Sugrue, Maryborough, Qld.
John Trigge, South Penrith

Editorial

As you can see by the list of new members elsewhere in this issue, A.C.E.(N.S.W.) is really growing. And growing strong. But we still want more members, not because we need them, but because they need us. We want to be able to truly represent all the ATARI owners in N.S.W. (and Australia for that matter). It's not until we are a strong body that we'll be able to exert any influence over the stuffed marketing decisions of you-know-who down in Melbourne. So how about it? If you're reading this magazine and you're not a member, then send your money in today! The cost of membership is more than covered by the programs in this issue alone. If these were available commercially, you'd be up for well over $50, whereas first time membership is only $30. If you're already a member, then make sure you tell everybody else about A.C.E.(N.S.W.). And make sure your local dealer knows about us too - bulk membership forms are available on request. One more thing ... participate! There's some great stuff in store for the future - all of which will be revealed in due course. Only by actively engaging in our various upcoming projects will you be able to get the most benefit out of <u>your</u> users' group.

As promised, this issue covers everything you need to know to get started with the extra graphics modes of the GTIA chip. And demos? Boy, we've got demos that'll blow your mind! Flex your fingers and dust off the old keyboard and get ready for an assault on the senses like you've never experienced before. I challenge <u>anybody</u> with <u>any</u> computer under say $5,000 to match these demos in just a few lines of BASIC code! Whatever you do, make sure you enter all these demos. Your ATARI will never be the same.

You may have noticed that INSIDE INFO has been growing. It now has smaller margins, more compact printing and more pages. All in all, more value for your money. I've also been placing more emphasis on programs, as this seems to be what people want. Notice also that BASIC listings are now given in "screen widths", i.e. 38 columns wide, so that what you see in the listing is what you should get on the screen - a great aid in debugging. The first issue had only one substantial program listing. The second issue increased to 5. This issue has a whopping 20 programs (or thereabouts) in BASIC and our first assembly language listings as well. Let's keep it up! We've already got the best magazine in Australia - let's make it the best in the world! If you haven't written anything yet, then now's the time to start. Don't forget that next month is games issue, so send in your favourite games. Articles should be addressed to the Editor, care of the usual address, or brought to the next meeting.

Lastly, please don't forget the A.G.M. in November. This is the most important meeting of the year, as we will elect our first committee. Once we've got a committee, things should really go ahead like a house on fire. If you're a member, make every effort to come to this meeting and exercise your right to vote. Or better still, offer yourself as a possible committee member.

- Garry Francis

"How's the programming going?"

GTIA

GRAPHICS 9, 10 & 11 — Part 1

by Garry Francis

INTRODUCTION

From January this year, all ATARI's sold in the U.S.A. have had a GTIA chip installed rather than the original CTIA. This has caused a surge of articles in all the popular computer magazines which have tried to explain the differences between the 2 chips. In the process, a great deal of confusion has arisen in the minds of ATARI owners in Australia . "Does my computer have CTIA or GTIA?" Unfortunately, due to the total lack of support from Futuretronics (the ATARI distributor in Australia), most owners do not even realise that GTIA exists, so lets get one point clear right from the start. Every ATARI 400 or 800 ever sold in Australia has got a GTIA chip installed.

GTIA would have to be the greatest invention of mankind since the ring-pull beer can. It gives the ATARI even more power ... power previously unheard of from any computer under about $100,000. A brief history of GTIA was printed in "Odds & Sods" in Issue No.1 and a short description of its functions was given in "What Makes It Tick?" in Issue No.2. Generally speaking, the GTIA chip functions exactly the same as CTIA except for the provision of 3 extra graphics modes known as GRAPHICS 9, 10 and 11. It is these 3 modes that we will explore in this issue and believe me, they are worth knowing about.

When I began to write this article, I quickly realised that I'd be "spittin' in the wind" if I expected everybody to understand what I was talking about, as most of our members are newcomers to computers. Therefore, I've decided to break the article down into 2 parts. This first part explains everything you need to know in simple terms in order to actually use GRAPHICS 9, 10 and 11. Where some general background in GRAPHICS modes is required, it is given. Even so, a great deal of what follows requires prior knowledge of other topics such as binary numbers, hardware registers, the Operating System and so on. We can't afford the space to explain all these things, but I'll do the best I can. If something seems a bit too technical, just skip over it and go onto the next section. (If you'd like to see articles on any of these topics in future issues, then let us know.) All addresses or memory locations are given in decimal with their hexadecimal equivalent given in square brackets.

Part 2 will be published some time in the future and will cover how GRAPHICS 9, 10 and 11 work. I'll also include some tips like how to get a text window in these modes. So without further ado, let's look at GRAPHICS 9, 10 and 11.

RESOLUTION AND PIXEL SIZE

All the bit-mapped graphics modes on the ATARI (as distinct from text/character modes) consist of a number of points on the screen. Each of these points is referred to as a pixel (which is derived from the words "picture element"). The BASIC Reference Manual tells us that there are 6 bit-mapped modes available, i.e. GRAPHICS 3 to GRAPHICS 8. Where these modes differ is in the size of the pixel and the number of colours available (refer to Tables 9.1 and 9.5 of your BASIC Reference Manual).

When talking about bit-mapped modes, you will often hear the term resolution. Resolution refers to the fineness of detail available. A display with large pixels can only show a low degree of detail and is therefore called low resolution. GRAPHICS 3 is low resolution. A display with very small pixels can show a high degree of detail and is therefore called high resolution. The maximum resolution on the ATARI is 320 pixels across the screen by 192 down the screen. This is available from BASIC in GRAPHICS 8+16. In order to compare the size of pixels in the various modes, I hereby announce a GRAPHICS 8 pixel to be 1 dot. (Technically, a dot is half a colour clock wide by one scan line high.) The size of a pixel in GRAPHICS 3 is therefore 8 dots by 8 dots, in GRAPHICS 4 and 5 it is 4 dots by 4 dots, in GRAPHICS 6 and 7 it is 2 dots by 2 dots and in GRAPHICS 8 it is 1 dot by 1 dot. There are actually 2 other modes available (but not accessible with the GRAPHICS statement) whose pixel size is 2 dots wide by 1 dot high.

This brings us to GRAPHICS 9, 10 and 11, which all have the same resolution and pixel size. Their resolution is 80 pixels across the screen by 192 pixels down (there is no text window). Any pixel may be referenced by giving its horizontal and vertical (X and Y) co-ordinates. X co-ordinates are numbered 0 to 79 and Y co-ordinates are numbered 0 to 191. The size of each pixel is 4 dots wide by 1 dot high (i.e. 2 colour clocks wide by 1 scan line high). In other words, each pixel is a tiny rectangle with the same width as a pixel in GRAPHICS 4 or 5, but the same height as a pixel in GRAPHICS 8.

COLOUR

The ATARI uses 2 parameters to specify a colour. These are hue and luminance. Hue is the proportion of the red, green and blue constituents of a colour and luminance is its brightness. The combination of hue and luminance define the colour. There are 9 hardware registers which may each store colour information (refer to the Table). As these are write-only registers, they each have a shadow register in page 2 of RAM. During each vertical blank, the Operating System reads the values in the shadow registers, does colour shifting if the attract flag (ATRACT [$4D]) is negative and writes the resultant values into the corresponding hardware register. In BASIC, the SETCOLOR statement may be used to change the colours in 5 of the shadow registers. 4 of these are referred to as playfield registers and the remaining one is the background register. These are located from address 708 to 712 [$02C4 to $02C8] and correspond to SETCOLOR 0 to SETCOLOR 4 respectively (again refer to the Table).

TABLE OF COLOUR REGISTERS

FUNCTION	O.S. SHADOW REGISTER			HARDWARE REGISTER			EQUIVALENT IN BASIC
	LABEL	DECIMAL ADDRESS	HEX ADDRESS	LABEL	DECIMAL ADDRESS	HEX ADDRESS	
PLAYER 0	PCOLR0	704	2C0	COLPM0	53266	D012	-
PLAYER 1	PCOLR1	705	2C1	COLPM1	53267	D013	-
PLAYER 2	PCOLR2	706	2C2	COLPM2	53268	D014	-
PLAYER 3	PCOLR3	707	2C3	COLPM3	53269	D015	-
PLAYFIELD 0	COLOR0	708	2C4	COLPF0	53270	D016	SETCOLOR 0
PLAYFIELD 1	COLOR1	709	2C5	COLPF1	53271	D017	SETCOLOR 1
PLAYFIELD 2	COLOR2	710	2C6	COLPF2	53272	D018	SETCOLOR 2
PLAYFIELD 3	COLOR3	711	2C7	COLPF3	53273	D019	SETCOLOR 3
BACKGROUND	COLOR4	712	2C8	COLBK	53274	D01A	SETCOLOR 4

An alternative method of changing the contents of these registers is to POKE them directly. The value to POKE is easily calculated from the formula:

COLOUR=HUE*16+LUMINANCE

where HUE and LUMINANCE are the second and third parameters of the normal SETCOLOR statement and the same value limitations apply (i.e. HUE is from 0 to 15 and LUMINANCE is an even number from 0 to 14). For example, to get a light green background, you would normally use SETCOLOR 4,11,6. This could also be done by using POKE 712,182 (11*16+6=182).

The shadow registers PCOLR0 to PCOLR3 from address 704 to 707 [$02C0 to $02C3] are used for player-missile graphics. As P-M graphics are not directly supported by BASIC, you cannot use the SETCOLOR statement to change the colours in these registers, so the POKE method must be used instead.

GRAPHICS 9

GRAPHICS 9 allows up to 16 different luminances of the same hue. The hue is specified using SETCOLOR 4,HUE,0. Hue may be any value from 0 to 15, but the luminance parameter must be 0 (this will be explained in Part 2). Once this is done, the COLOR statement is used to switch between luminances, just as it is normally used to switch between colour registers. The parameter in the COLOR statement may be anything from 0 to 15 with 0 being the darkest luminance (the same as the background) and 15 being the brightest. Study Programs 1 and 2 to see this in effect. Program 1 draws 16 vertical bars across the screen to show the range of luminances available. Program 2 draws horizontal lines which illustrate the fine degree of shading available. In both programs, once the screen is filled, you may press the < key to decrease the hue value or press > to increase the hue value.

Note that GRAPHICS 9 is the only mode that allows 16 luminances - you are normally restricted to 8. What this means is that if you had a few display list interrupts, you could easily have 256 colours on the screen at once!!

What's wrong? Don't you believe me?

Whenever someone tells you that the ATARI can do something better than all the other personal computers on the market, you'd better believe it! Run Program 3 and you'll see what I mean. The next time your friend with the Apple (or VIC or Tandy Colour Computer or whatever) criticises your ATURKEY 400 for only being able to display 4 colours on the screen at once, show him this program and ask him to match it! Note that this program is particularly handy as a "palette" for selecting colours, as every possible colour is displayed on the screen at once. I won't elaborate on how the program works as the REMarks pretty well cover it. The POKE in line 10 is a precautionary measure to ensure that Display List Interrupts are disabled (but the Vertical Blank is still enabled) while everything is being set up. The USR routine is simply a delay to ensure that DLI's are not enabled until during the Vertical Blank. The assembled source code for the 2 machine language routines is given for anybody interested.

GRAPHICS 10

GRAPHICS 10 allows 9 colours in any hue/luminance combination by utilising the full 9 colour registers. It is the only mode that does so. The COLOR statement is used to switch between colour registers just as usual, but the parameter may be any value from 0 to 8. These correspond to the shadow registers from address 704 to 712 respectively. Note that the background colour is now stored at address 704, whereas most other modes use address 712. Program 4 is a simple illustration of how to display 9 colours in GRAPHICS 10. The DATA statements may be changed to allow any colours you desire.

GRAPHICS 11

GRAPHICS 11 is very similar to GRAPHICS 9 except that it allows up to 16 different hues of the same luminance. The luminance must be specified using SETCOLOR 4,0,LUM. LUM may be any even number from 0 to 14, but the hue parameter must be 0. The COLOR statement is used to switch between hues using any value from 0 to 15 as the parameter. These correspond to the hues in Table 9.3 of the BASIC Reference Manual. Note however, that the background (and hence COLOR 0) is always black regardless of the luminance setting. Programs 5 and 6 are the equivalent of Programs 1 and 2 to show how to use GRAPHICS 11.

```
1 REM ###############################
2 REM #          PROGRAM 1          #
3 REM #   GRAPHICS 9 COLOUR SCALE   #
4 REM #       by Garry Francis      #
5 REM # Published by Atari Computer #
6 REM #    Enthusiasts (N.S.W.)     #
7 REM #         October 1982        #
8 REM ###############################
10 GRAPHICS 0:POSITION 12,1:? "GRAPHIC
S 9 DEMO":? :? "When colour scale is d
rawn..."
20 ? :? "  Press < to decrease HUE":?
:? "  Press > to increase HUE":?
30 TRAP 170:? :? "Enter HUE (0-15)";:I
NPUT HUE
40 IF HUE<0 OR HUE>15 OR HUE<>INT(HUE)
 THEN 170
50 GRAPHICS 9:SETCOLOR 4,HUE,0
60 FOR C=1 TO 15:COLOR C
70 FOR X=C*5 TO C*5+4:PLOT X,0:DRAWTO
X,191:NEXT X
80 NEXT C
90 POKE 764,255
100 A=PEEK(764):IF A=54 THEN 130
110 IF A=55 THEN 150
120 GOTO 100
130 HUE=HUE-1:IF HUE<0 THEN HUE=15
140 SETCOLOR 4,HUE,0:GOTO 90
150 HUE=HUE+1:IF HUE>15 THEN HUE=0
160 SETCOLOR 4,HUE,0:GOTO 90
170 ? :? "ILLEGAL ENTRY. PLEASE TRY AG
AIN.":GOTO 30
```

```
1 REM ##############################
2 REM #         PROGRAM 2          #
3 REM #     GRAPHICS 9 SHADING     #
4 REM #      by Garry Francis      #
5 REM # Published by Atari Computer #
6 REM #    Enthusiasts (N.S.W.)    #
7 REM #        October 1982        #
8 REM ##############################
10 GRAPHICS 0:POSITION 12,1:? "GRAPHIC
S 9 DEMO":? :? "When picture is drawn.
.."
20 ? :? "  Press ≤ to decrease HUE":?
:? "  Press ≥ to increase HUE":?
30 TRAP 170:? :? "Enter HUE (0-15)";:I
NPUT HUE
40 IF HUE<0 OR HUE>15 OR HUE<>INT(HUE)
 THEN 170
50 GRAPHICS 9:SETCOLOR 4,HUE,0:C=0
60 FOR Y=0 TO 191:COLOR C:PLOT 0,Y:DRA
WTO 79,Y
70 C=C+1:IF C>15 THEN C=0
80 NEXT Y
90 POKE 764,255
100 A=PEEK(764):IF A=54 THEN 130
110 IF A=55 THEN 150
120 GOTO 100
130 HUE=HUE-1:IF HUE<0 THEN HUE=15
140 SETCOLOR 4,HUE,0:GOTO 90
150 HUE=HUE+1:IF HUE>15 THEN HUE=0
160 SETCOLOR 4,HUE,0:GOTO 90
170 ? :? "ILLEGAL ENTRY. PLEASE TRY AG
AIN.":GOTO 30
```

```
1 REM ##############################
2 REM #         PROGRAM 4          #
3 REM #     GRAPHICS 10 DEMO       #
4 REM #      by Garry Francis      #
5 REM # Published by Atari Computer #
6 REM #    Enthusiasts (N.S.W.)    #
7 REM #        October 1982        #
8 REM ##############################
10 GRAPHICS 10:POKE 704,14
20 FOR I=1 TO 8:READ A:POKE 704+I,A:CO
LOR I
30 FOR X=I*10-10 TO I*10-1:PLOT X,0:DR
AWTO X,191:NEXT X:NEXT I
40 GOTO 40
50 DATA 54,40,28,182,116,68,60,8
```

```
1 REM ##############################
2 REM #         PROGRAM 5          #
3 REM #  GRAPHICS 11 COLOUR SCALE  #
4 REM #      by Garry Francis      #
5 REM # Published by Atari Computer #
6 REM #    Enthusiasts (N.S.W.)    #
7 REM #        October 1982        #
8 REM ##############################
10 GRAPHICS 0:POSITION 12,1:? "GRAPHIC
S 11 DEMO":? :? "When colour scale is
drawn..."
20 ? :? "  Press ≤ to decrease LUMINAN
CE":? :? "  Press ≥ to increase LUMINA
NCE":?
30 TRAP 170:? :? "Enter LUMINANCE (eve
n No. 0-14)";:INPUT LUM
40 IF LUM<0 OR LUM>14 OR LUM<>INT(LUM)
 OR LUM/2<>INT(LUM/2) THEN 170
50 GRAPHICS 11:SETCOLOR 4,0,LUM
60 FOR C=1 TO 15:COLOR C
70 FOR X=C*5 TO C*5+4:PLOT X,0:DRAWTO
X,191:NEXT X
80 NEXT C
90 POKE 764,255
100 A=PEEK(764):IF A=54 THEN 130
110 IF A=55 THEN 150
120 GOTO 100
130 LUM=LUM-2:IF LUM<0 THEN LUM=14
140 SETCOLOR 4,0,LUM:GOTO 90
150 LUM=LUM+2:IF LUM>14 THEN LUM=0
160 SETCOLOR 4,0,LUM:GOTO 90
170 ? :? "ILLEGAL ENTRY. PLEASE TRY AG
AIN.":GOTO 30
```

```
1 REM ##############################
2 REM #         PROGRAM 6          #
3 REM #    GRAPHICS 11 SHADING     #
4 REM #      by Garry Francis      #
5 REM # Published by Atari Computer #
6 REM #    Enthusiasts (N.S.W.)    #
7 REM #        October 1982        #
8 REM ##############################
10 GRAPHICS 0:POSITION 12,1:? "GRAPHIC
S 11 DEMO":? :? "When picture is drawn
..."
20 ? :? "  Press ≤ to decrease LUMINAN
CE":? :? "  Press ≥ to increase LUMINA
NCE":?
30 TRAP 170:? :? "Enter LUMINANCE (eve
n No. 0-14)";:INPUT LUM
40 IF LUM<0 OR LUM>14 OR LUM<>INT(LUM)
 OR LUM/2<>INT(LUM/2) THEN 170
50 GRAPHICS 11:SETCOLOR 4,0,LUM:C=0
60 FOR Y=0 TO 191:COLOR C:PLOT 0,Y:DRA
WTO 79,Y
70 C=C+1:IF C>15 THEN C=0
80 NEXT Y
90 POKE 764,255
100 A=PEEK(764):IF A=54 THEN 130
110 IF A=55 THEN 150
120 GOTO 100
130 LUM=LUM-2:IF LUM<0 THEN LUM=14
140 SETCOLOR 4,0,LUM:GOTO 90
150 LUM=LUM+1:IF LUM>14 THEN LUM=0
160 SETCOLOR 4,0,LUM:GOTO 90
170 ? :? "ILLEGAL ENTRY. PLEASE TRY AG
AIN.":GOTO 30
```

```
1 REM ##############################
2 REM #          PROGRAM 3          #
3 REM #  256 COLOURS IN GRAPHICS 9  #
4 REM #      by Garry Francis       #
5 REM # Published by Atari Computer #
6 REM #    Enthusiasts (N.S.W.)     #
7 REM #        October 1982         #
8 REM ##############################
9 REM *** SET DLI BIT FOR EVERY 12th L
INE OF GR.9 DISPLAY LIST ***
10 GRAPHICS 9:POKE 54286,64:DL=PEEK(56
0)+256*PEEK(561)
20 FOR I=DL+16 TO DL+88 STEP 12:POKE I
,143:NEXT I
30 FOR I=DL+102 TO DL+186 STEP 12:POKE
 I,143:NEXT I
39 REM *** STORE DLI SERVICE ROUTINE I
N PAGE 6 ***
40 FOR I=1536 TO 1560:READ A:POKE I,A:
NEXT I:POKE 512,0:POKE 513,6
50 DATA 72,24,173,24,6,105,16,141,10,2
12,141,26,208,201,240,208,2,169,0,141,
24,6,104,64,0
59 REM *** STORE USR ROUTINE IN PAGE 6
 ***
60 FOR I=1561 TO 1578:READ A:POKE I,A:
NEXT I
70 DATA 104,169,0,141,24,6,165,20,197,
20,240,252,169,192,141,14,212,96
79 REM *** DRAW VERTICAL BARS IN 16 LU
MINANCES ***
80 FOR C=1 TO 15:COLOR C:FOR X=C*5 TO
C*5+4:PLOT X,0:DRAWTO X,191:NEXT X:NEX
T C
89 REM *** WAIT FOR VERTICAL BLANK BEF
ORE ENABLING DLI'S ***
90 X=USR(1561)
100 GOTO 100
```

```
                  0100 ;###############################
                  0110 ;#   SOURCE CODE FOR MACHINE   #
                  0120 ;# LANGUAGE STUFF IN PROGRAM 3 #
                  0130 ;#      by Garry Francis       #
                  0140 ;# Published by Atari Computer #
                  0150 ;#    Enthusiasts (N.S.W.)     #
                  0160 ;#        October 1982         #
                  0170 ;###############################
                  0180 ;
                  0190 ;DLI ROUTINE TO GET 256 COLOURS
                  0200 ;ON THE SCREEN IN GRAPHICS 9
                  0210 ;
D01A              0220 COLBK  =    $D01A
D40A              0230 WSYNC  =    $D40A
0000              0240        *=   $0600
0600 48           0250        PHA              ;Save accumulator
0601 18           0260        CLC              ;Prepare for addition
0602 AD1806       0270        LDA  COUNTR      ;Get current colour
0605 6910         0280        ADC  #$10        ;Increment left nybble (i.e. hue)
0607 8D0AD4       0290        STA  WSYNC       ;Wait for horizontal blank
060A 8D1AD0       0300        STA  COLBK       ;Store colour in background register
060D C9F0         0310        CMP  #$F0        ;Is hue at maximum?
060F D002         0320        BNE  END         ;No...leave colour as is
0611 A900         0330        LDA  #0          ;Yes...reset colour to zero
0613 8D1806       0340 END    STA  COUNTR      ;Save new colour in RAM counter
0616 68           0350        PLA              ;Restore accumulator
0617 40           0360        RTI              ;Return from interrupt
0618 00           0370 COUNTR .BYTE 0          ;RAM counter to hold colour value
                  0380 ;
                  0390 ;USR FUNCTION TO WAIT FOR VBLANK
                  0400 ;BEFORE ENABLING DLI
                  0410 ;
0014              0420 RTCLOK =    $14
D40E              0430 NMIEN  =    $D40E
0619 68           0440        PLA              ;Pull no. of arguments off stack
061A A900         0450        LDA  #0          ;Zero the...
061C 8D1806       0460        STA  COUNTR      ;...RAM counter
061F A514         0470        LDA  RTCLOK      ;Get LSB of real time clock
0621 C514         0480 WAIT   CMP  RTCLOK      ;Has it changed?
0623 F0FC         0490        BEQ  WAIT        ;No...try again
0625 A9C0         0500        LDA  #$C0        ;Yes...therefore must be in vertical blank
0627 8D0ED4       0510        STA  NMIEN       ;Enable DLI
062A 60           0520        RTS              ;Return to BASIC
062B              0530        .END
```

RAMSEY'S RAMBLINGS

Keyboard Organ, String Graphics & GTIA Demo

by Gregg Ramsey

This month I have some reasonable programs to share with you, but I may have had some better ones if I'd been able to find some more time to use my computer. I recently purchased a Disk Drive and would like to mention that it has been an investment second only to the ATARI itself. There are many reasons for this. The first and most obvious reason is the speed. I have "Rescue at Rigel" on tape and had stopped playing it since it took a whopping seven minutes to load (not to mention finding and positioning it). I made a backup copy of it on disk and found that it took a mere eighteen seconds to load. Another plus for the disk is that it keeps its own directory - meaning that I don't have to fiddle about with loose scraps of paper any more. One more great feature is that when I'm writing long programs (such as these articles), I can constantly SAVE it. When I was writing the last article with the cassette, the computer took a sicky a few times and I lost everything. Well, my worrying is now much less and a whole new range of software is now open to me - including other languages. So I just want to say that a Disk Drive is really worth the money even if it sounds like a lot.

LISTING 1: ORGAN

Firstly, I would like to have a little pick at the BASIC Reference Manual. In case you hadn't noticed, it has some demo programs in Appendix H. One in particular on page H-15 is called "Type-a-Tune". Well, this isn't a bad gripe, it's just that I don't think they've done it in a very efficient way. They chose to define two arrays - one to hold the values of PEEK(764) and the other to hold the notes. It holds 12 notes so that every time you press a key, it goes through a FOR...NEXT loop of order 12 to find the relevant value. My method uses one array with the element numbers corresponding to the PEEK(764) values and containing the pitch required should that number arise. This means that it's just as fast to have a 3 key piano (in BASIC) as it is to have 256 keys. It also means that extra notes can easily be added. Instead of a FOR...NEXT loop dependent on the number of notes required, this method only needs the statement:

```
SOUND 0,TUNE(PEEK(764)),10,8
```

where TUNE() is the array holding the notes. In my program, there are two full octaves and all their sharps and flats and a neat little spruce-up feature called RESONATE, whereby pressing the SHIFT key with the note key adds a second voice offset by a pitch value of 1. Make sure you type the underlined characters in INVERSE video. Another feature is that the program prints the pitch value of the note currently being played. This is so that you can familiarise yourself with these numbers and eventually learn them so that you won't need to look up those tables.

I have translated some tunes into the "Organ language" to get you started. It is initially quite difficult to read, but after a little while, I'm sure you'll get the hang of it. Just press each of the keys shown in turn. The timing will come to you naturally. A slash indicates a pause.

Jingle Bells

O O O/O O O/O - U I O/P P P/P P O O/O O I I O I -
O O O/O O O/O - U I O/P P P/P P O O/O - - P I U

Mary Had A Little Lamb

W Q TAB Q W W W/Q Q Q/W R R
W Q TAB Q W W W/W Q Q W Q TAB

Knuckles

4 5 6/8 8/4 5 6/8 8/6 5 4/2 2/6 5 4/2 2
4 5 6/8 8/4 5 6/8 8/U 8 9 P < < <

Close Encounters (make sure you do this with RESONATE on)

```
I O U TAB R
```

Star Wars

```
U -/P O I LOWR -/P O I LOWR -/P O P I
U -/P O I LOWR -/P O I LOWR -/P O P I
R T/P O I U U I O I T Y
R T/P O I U - I
R T/P O I U U I O I T Y
- - LOWR BS > - P 9 I U/- - - -/- - - -/(repeat)
```

Program Description

Line 10: Goes to initialisation subroutine.

Line 20: Location 53775 is a hardware register called SKSTAT (Serial Port - Keyboard Status [$D20F]). It returns 255 if no key is pressed, 251 if any key except SHIFT is pressed, 247 if only SHIFT is pressed and 243 if SHIFT and another key are pressed together.

Line 30: Sets up a variable B containing the value of memory location 764 and then plays the note with a PITCH value obtained from the array element TUNE where the element number is B (wow!).

Line 40: Makes the sound and prints its pitch value.

Lines 50-60: The RESONATE part. If the SHIFT key is pressed, sound register 1 plays a pitch of one greater than register 0.

Line 70: Turn sounds off if no key is pressed.

Line 100: Disable break key so that you don't accidentally bomb the program when pressing keys near it.

Lines 110-120: Set up colours and turn cursor off.

Line 140: DIMension the array to hold the pitch values. For example, if we want the TAB key to play a C (PITCH=243), then we assign 243 to the element whose number is the same as the number returned by PEEK(764) when the TAB key is pressed (double wow!).

Line 150: Set all the elements to zero.

Line 160: Read the element numbers and the corresponding PITCH values and put them in. Notice that for each PITCH, there are two array element numbers. This is for the benefit of the RESONATE function because PEEK(764) returns a different value when the SHIFT key is being held down.

Lines 200-280: Print a picture of the keyboard on the screen. The keyboard is laid out like a piano keyboard except that the last C had to go on the CAPS/LOWR key. Tough luck.

Lines 300-320: The data. Make sure you get them all right and don't leave spaces as it will be interpreted as string data and you'll get an ERROR - 8.

LISTING 2: STRING GRAPHICS

The next program I want to share with you has a neat technique for doing intricate pictures where most of the dots next to each other are a different colour and the PLOT and DRAWTO commands would take up too much memory. Have you ever tried something like... PRINT #6;"HELLO" in say GRAPHICS 5. Well if you have, you would have noticed that it would have obeyed you like a slave and printed them there except that they were dots (of course) and not letters. If you do a little experimenting, you will find that if you PRINT an A, a dot coloured by colour register 0 appears, a B is coloured by register 1, a C by register 2 and D by register 4 (the background colour). The program DIMensions one string variable A$ to hold the line of text to be printed. There is a loop using ATARI's neat feature of enabling it to GOSUB to expressions as well as numbers. It will GOSUB or GOTO anything that evaluates to an integer. Well, this expression GOSUBs lines to fetch values for A$, starting at line 1000 and incrementing by 10 to 1010 then 1020 and so on. This program draws the ATARI logo in 4 GRAPHICS modes. Notice also that you can press the number of the GRAPHICS mode that you want while the program is running and it will automatically change over. Using this technique, I made a program that draws a very detailed picture of Snoopy and his little friend Woodstock in GRAPHICS 7. It filled up a lot of the screen but is too large for this article.

Program Description

Line 10: DIMension the string to hold the pixel data.

Line 20: Select GRAPHICS mode from the value of GR (this is set in lines 600-630), then colour everything black.

Lines 30-60: Prints the picture to the screen with the top left hand corner positioned at (X,Y).

Line 70: Clear the character buffer. This is necessary because in the next loop, I check to see if a key is pressed. If I didn't clear it, the key last pressed would be ready and waiting and there would be terrible confusion.

Lines 80-90: The final loop. This rotates the colours of the picture to give it some character. It also checks to see if a key has been pressed. If it has been, then the program goes to check which key it was from line 590 onwards and determines a course of action. Note that REG is a variable containing the colour register to POKE. This stays at 708 unless GRAPHICS 8 is selected in which case it changes to 709. There is also a delay loop so that the colours dont flash past too quickly.

Lines 500-580: Prints instructions.

Line 590: Opens the keyboard as an input device and sets REG to its default value of 708.

Lines 600-630: Scans the keyboard until one key is pressed and gets its ATASCII value. Sets up variables for the GRAPHICS mode selected and hands control over to the main program at Line 20.

Line 640: If it gets to here, the wrong key was pressed so go back and get another key at line 600.

Lines 1000-1110: The data for A$. Feel free to put whatever picture you desire in here and see what it looks like in the different GRAPHICS modes.

LISTING 3: GTIA DEMO

The basic theme of this month's issue is the GTIA chip. Since its secrets have been revealed elsewhere and there are lots of demos, I didn't think that I could add much light to the topic, although I did dig up one demo involving all three of GTIA's extra GRAPHICS modes. This program shows virtually infinite variations of colours and shapes but I must add that I can't claim fame to the guts of the program. I saw it at Computer Wave and still don't know the author. I added the instructions and the optional GRAPHICS modes. The part that puts all the colours onto the screen so nicely is the COLOR statement in line 120. A neat feature is that you can view pictures in GRAPHICS 8, 9, 10 and 11. In GRAPHICS 8, you will notice some nice artifacting (see Issue No. 1), as the points are individually plotted next to each other. When the picture is being drawn you can press CTRL 1 to stop and start or the START key to get back to the instructions. When you start the program, you will be asked to enter 2 numbers (seeds). The first number must be greater than 0 and less than 13000. Each number represents a "page" of the picture, i.e. if you enter 1 for the start and 10 for the end, then you will get 10 pictures. The second number must be greater than or equal to the first so if you only want to see picture number 20, type 20 for the beginning and end. The program then asks you for a GRAPHICS mode. You can select anything from GRAPHICS 8 to 11.

Program Description

Line 100: Initialise. Go and get the GRAPHICS mode we'll be working in and the start and finish seeds.

Line 110: Checks to see if the START key is pressed. If it is then RUN the program again.

Line 120: The complicated COLOR statement. Without this, the program would be nothing. Try changing the expression around to get even more patterns.

Lines 130-140: The almost equally complicated plotting part. It plots 8 points at a time starting from each corner to get symmetry.

Line 150: Finish off and then click the internal speaker.

Lines 160-170: The end loop. You can either watch the picture until your heart's content or press START to go back to the instructions and have another try or look at the same pictue in a different GRAPHICS mode. Strangely enough there is an easier way to observe the same picture in a different GRAPHICS mode. If your picture was in GRAPHICS 10 and you want to see what it looks like in GRAPHICS 9, then press the BREAK key and add the following 2 lines to the program. Do not RUN it, but enter GOTO 2000 in direct mode.

```
2000 ? CHR$(125):GRAPHICS 9+32
```

```
2010 GOTO 2010
```

The ? CHR$(125) is there to clear the muck at the bottom since GRAPHICS 0 and 9 share the same display memory.

Lines 1000-1100: These are the lines that get the input. Notice the use of the handy TRAP statement to check for idiot input.

Line 1110: The variable D is only used in GRAPHICS 8 to offset the picture so that it becomes centred. If GRAPHICS 8 is selected then start the picture 120 pixels to the right of the origin. This value gets added to each X-point plotted. CONSOL is ATARI's label for the hardware register that reads the console keys.

Line 1120: Poke the colours in and put us into the desired GRAPHICS mode.

That concludes my article for this month. I hope that you've got a little something out of it, because I certainly did in the process of writing it. Next issue is FUN issue because the theme is GAMES. I hope to have a game or two. If you've written any games, then I urge you to submit them to the Editor. They may contain a programming technique or an idea that not many people are familiar with, so get 'em in (or start writing).

This is Gregg Ramsey saying good bye for now!

```
1 REM ##############################
2 REM #       LISTING 1:  ORGAN      #
3 REM #        by Gregg Ramsey       #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)     #
6 REM #         October 1982        #
7 REM ##############################
10 GRAPHICS 0:GOSUB 100:REM INITIALISE
19 REM xxx THIS IS THE MAIN LOOP xxx
20 A=PEEK(SKSTAT):IF A<>243 AND A<>251
 THEN 20
30 B=PEEK(764):IF TUNE(B)=0 THEN 20
40 SOUND 0,TUNE(B),10,8:POSITION 22,12
:? TUNE(B);"  "
50 IF PEEK(SKSTAT)=243 THEN SOUND 1,TU
NE(B)+1,10,8:GOTO 30
60 IF PEEK(SKSTAT)=251 THEN SOUND 1,0,
0,0:GOTO 30
70 SOUND 0,0,0,0:SOUND 1,0,0,0:POSITIO
N 22,12:? "0  ":GOTO 20
99 REM xxx INITIALISATION xxx
100 POKE 16,64:POKE 53774,119:REM DISA
BLE BREAK KEY
110 POKE 709,2:POKE 710,12:POKE 712,56
:REM POKE COLOURS IN
120 SKSTAT=53775:POKE 752,1
130 POSITION 13,3:? "Setting up..."
140 DIM TUNE(255):REM HOLDS NOTES
150 FOR T=0 TO 255:TUNE(T)=0:NEXT T:RE
M SET ALL NOTES TO PITCH=0
160 READ K1,K2,N:IF K1<>-1 THEN TUNE(K
1)=N:TUNE(K2)=N:GOTO 160
199 REM xxx PRINT TITLE AND PIANO KEYS
 xxx
200 POSITION 13,3:? "Organ  Player"
210 POSITION 7,6:? "1 2   4 5 6   8 9
  < > BS"
220 POSITION 4,8:? "TAB Q W E R T Y U
I O P - = RET"
230 POSITION 5,9:? "C  D E F G A B C D
 E F G A  B"
240 POSITION 31,11:? "LOWR"
250 POSITION 16,12:? "PITCH=0"
260 POSITION 32,12:? "C"
270 POSITION 4,14:? "SHIFT":POSITION 3
1,14:? "SHIFT"
280 POSITION 4,15:? "RESONATE":POSITIO
N 28,15:? "RESONATE"
290 RETURN
299 REM xxx DATA FOR KEYS & NOTES xxx
300 DATA 31,95,230,30,94,204,24,88,173
,29,93,153,27,91,136,53,117,114,48,112
,102,54,118,85,55,119,76,52,116,68
310 DATA 44,108,243,47,111,217,46,110,
193,42,106,182,40,104,162,45,109,144,4
3,107,128,11,75,121,13,77,108
320 DATA 8,72,96,10,74,91,14,78,81,15,
79,72,12,76,64,60,124,60,-1,0,0
```

...STOP PRESS...

Unfortunately, our regular columns (i.e. Copyright Law and Odds & Sods) and a couple of articles were squeezed out of this issue due to the sheer bulk of the material on GTIA. They will now appear in the next issue. My apologies to the authors concerned.

– The Editor

```
1 REM ##############################
2 REM # LISTING 2:  STRING GRAPHICS #
3 REM #        by Gregg Ramsey       #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)     #
6 REM #        October 1982         #
7 REM ##############################
10 DIM A$(16):GOTO 500:REM INITIALISE
20 CLOSE #1:GRAPHICS GR+16:SETCOLOR 0,
0,0:SETCOLOR 1,0,0:SETCOLOR 2,0,0
30 FOR T=0 TO 11
40 GOSUB 1000+T*10
50 POSITION X,Y+T:? #6;A$
60 NEXT T
70 POKE 764,255:REM CLEAR KEYBOARD BUF
FER
80 FOR T=0 TO 255:IF PEEK(764)<>255 TH
EN 590:REM ROTATE COLOURS AND CHECK FO
R KEY PRESS
90 POKE REG,T:FOR U=0 TO 10:NEXT U:NEX
T T:GOTO 80
500 GRAPHICS 0:POKE 752,1:POSITION 10,
1:? "STRING GRAPHICS DEMO"
510 ? :? "Select a GRAPHICS mode..."
520 ? :? "  3 for GRAPHICS 3"
530 ? "  5 for GRAPHICS 5"
540 ? "  7 for GRAPHICS 7"
550 ? "  8 for GRAPHICS 8"
560 POSITION 2,16:? "When running..."
570 ? :? "  Press 3, 5, 7 or 8 to sele
ct":? "  another mode."
580 ? :? "  Press any other key to sto
p colours":? "  changing."
590 OPEN #1,4,0,"K:":REG=708:REM OPEN
KEYBOARD FOR INPUT
600 GET #1,K:IF K=51 THEN GR=3:X=12:Y=
5:GOTO 20
610 IF K=53 THEN GR=5:X=32:Y=16:GOTO 2
0
620 IF K=55 THEN GR=7:X=72:Y=38:GOTO 2
0
630 IF K=56 THEN GR=8:X=152:Y=82:REG=7
09:GOTO 20
640 GOTO 600
1000 A$="     A AA A     ":RETURN
1010 A$="     A AA A     ":RETURN
1020 A$="     A AA A     ":RETURN
1030 A$="     A AA A     ":RETURN
1040 A$="     A AA A     ":RETURN
1050 A$="    AA AA AA    ":RETURN
1060 A$="    AA AA AA    ":RETURN
1070 A$="   AA  AA  AA   ":RETURN
1080 A$="  AAA  AA  AAA  ":RETURN
1090 A$="AAAA   AA   AAAA":RETURN
1100 A$="AAAA   AA   AAAA":RETURN
1110 A$="AA     AA     AA":RETURN
```

```
1 REM ##############################
2 REM #    LISTING 3:  GTIA DEMO    #
3 REM #        by Gregg Ramsey       #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)     #
6 REM #        October 1982         #
7 REM ##############################
100 GOSUB 1000:FOR W=A TO B:FOR I=1 TO
 39:FOR J=0 TO 39:K=I+J
110 IF PEEK(CONSOL)=6 THEN RUN
120 COLOR J*3/(I+3)+I*W/8:REM COLOUR R
OUTINE (THE GUTS OF THE PROGRAM)
130 PLOT I+D,K*2:PLOT K+D,I*2:PLOT 80-
I+D,160-(K*2):PLOT 80-K+D,160-(I*2)
140 PLOT K+D,160-(I*2):PLOT 80-I+D,K*2
:PLOT I+D,160-(K*2):PLOT 80-K+D,I*2
150 NEXT J:NEXT I:NEXT W:POKE CONSOL,0
160 IF PEEK(CONSOL)=6 THEN RUN
170 GOTO 160
1000 GRAPHICS 0:DIM B$(37):B$="
                              ";REM 37
 BLANKS
1010 POSITION 15,1:? "GTIA DEMO"
1020 POSITION 2,18:? "When running..."
1030 ? :? "  Press CTRL 1 to freeze di
splay.":? :? "  Press START to get bac
k here."
1040 TRAP 1040:POSITION 2,3:? B$:POSIT
ION 2,3:? "Enter first seed (0-13000)"
;:INPUT A
1050 IF A<0 OR A>13000 OR A<>INT(A) TH
EN 1040
1060 IF A=13000 THEN B=A:GOTO 1090
1070 TRAP 1070:POSITION 2,5:? B$:POSIT
ION 2,5:? "Enter second seed (";A;"-13
000)";:INPUT B
1080 IF B<A OR B>13000 OR B<>INT(B) TH
EN 1070
1090 TRAP 1090:POSITION 2,7:? B$:POSIT
ION 2,7:? "Enter GRAPHICS mode (8-11)"
;:INPUT GR
1100 IF GR<8 OR GR>11 OR GR<>INT(GR) T
HEN 1090
1110 CONSOL=53279:D=0:IF GR=8 THEN D=1
20
1120 POKE 704,0:GRAPHICS GR+16:POKE 71
0,0:IF GR=10 THEN FOR I=705 TO 712:POK
E I,RND(0)*255:NEXT I
1130 RETURN
```

GTIA DEMOS

by Garry Francis

I searched high and low for some programs to illustrate the use of GRAPHICS 9, 10 and 11. Even though the search was more than successful, I found that everybody seemed to have a sub-set of the same group of programs. It is fairly certain that they originated somewhere at ATARI, but unfortunately, the author is completely unknown. I suspect that it may be Bill Carris, who has done several demos for the ATARI, but this is only a hunch. If the original author comes forward, we will certainly give him or her credit.

I would particularly like to thank the following sources for making these programs available:

- Creative Computing June 1982 for Beach Ball, Brass, Escape, Hypnosis, Psychedelic, Rolls, Sas, Water Melon and Whirl. (Note that I have renamed some of the programs to better describe their appearance.)
- Melbourne Atari Computer Enthusiasts for Brass, Cone, Dizzy, Escape, Grenhole, Hypnosis, Poles, Rolls, Stripes, Water Melon and Whirl.
- The guys in the Computer Department at Futuretronics for Beach Ball, Brass, Cone, Dizzy, Escape, Grenhole, Hypnosis, Poles, Psychedelic, Rolls, Sas, Stripes, Water Melon and Whirl.
- The Atari Connection Vol.2 No.2 for Hypnosis.

If you have any early versions of these programs, you might as well throw them away. The versions here have been totally re-written, but all retain the original concept. In most cases, you won't even recognise the code when compared to the original. The aim of the improvements was to:

- make the demonstrations aesthetically more pleasing, both in their final appearance and in the way they were drawn up (this included size, shape, symmetry, position on screen, colours used and elimination of flaws and "holes" in some patterns)
- speed up the execution
- provide more elegant coding, uniform use of variable names and overall improvement in understandability
- improve the algorithms used and eliminate several unnecessary steps
- cut down on memory requirements
- take up less room in the magazine

Unfortunately, INSIDE INFO has a deadline to meet, so I was under pressure to complete 20 or so programs within 2 weeks. Because of this, they could probably be tightened up even more, but that option is on you.

Several of the programs (viz. Dizzy, Escape, Grenhole, Hypnosis, Poles, Sas, Stripes and Whirl) use a short machine language subroutine. You may easily pass this off as being the reason for the spectacular displays. Not so! It is merely a simple colour rotation routine which could just as easily have been done in BASIC (in fact it is done this way in Beach Ball, Psychedelic and Rolls). In some cases, the colour rotation appears "jerky" if done in BASIC. This is when the machine language routine comes in handy. Just to prove how simple it is, I've given its assembled source code for those that are interested. Note that there is a "glitch" in the routine, whereby you will occasionally get a flash of black in one of the colour registers (it only lasts 1/50 second). This is probably related to bad syncronisation with the vertical blank.

I've put a lot of time into these programs, so let's not waste any more by talking about them. Screw your eyeballs in tight and prepare yourself for a psychedelic trip that no drug could provide! Don't leave any of the programs out thinking that they look the same, because I can assure you they are all totally different.

Notes

When you run Beach Ball, stand a long way back from the TV. If you feel the colours are too harsh or gaudy, change the DATA statement in line 50 to read:

50 DATA 24,56,72,88,136,168,184,216

Dizzy is particularly interesting, because it shows a simple method of providing text on a graphics screen. I lifted the idea from IRIDIS 1. It could easily be expanded to allow extra functions (such as fill, change colour or set a new origin) for drawing all sorts of pictures.

Most of the programs display something almost immediately, but Sas takes about 7 seconds to initialise.

```
               0100 ;############################
               0110 ;# COLOUR ROTATION SOURCE CODE #
               0120 ;#       by Garry Francis       #
               0130 ;# Published by Atari Computer #
               0140 ;#    Enthusiasts (N.S.W.)      #
               0150 ;#         October 1982         #
               0160 ;############################
               0170 ;
               0180 ;MACHINE LANGUAGE SUBROUTINE TO ROTATE ALL COLOURS (EXCEPT THE
               0190 ;BACKGROUND) IN GRAPHICS 10.
               0200 ;
               0210 ;CODE IS FULLY RELOCATABLE AND WOULD NORMALLY BE STORED IN A STRING
               0220 ;WHEN USING BASIC. STRING MUST BE DIMENSIONED TO 21, e.g. DIM A$(21).
               0230 ;
               0240 ;CALL FROM BASIC USING:
               0250 ;  X=USR(ADR(A$))
               0260 ;
0000           0270        *=   $600      ;Page 6 used to assemble program for convenience only
02C1           0280 PCOLR1 =    $02C1     ;Player 1 shadow register (decimal 705)
02C2           0290 PCOLR2 =    $02C2     ;Player 2 shadow register (decimal 706)
02C8           0300 COLOR4 =    $02C8     ;Background shadow register (decimal 712)
               0310 ;
0600 68        0320        PLA            ;Pull no. of arguments off stack
0601 A200      0330        LDX  #0        ;Initialise counter
0603 ACC102    0340        LDY  PCOLR1    ;Save for later
0606 BDC202    0350 LOOP   LDA  PCOLR2,X  ;Move colours down...
0609 9DC102    0360        STA  PCOLR1,X  ;one address
060C E8        0370        INX            ;Increment counter
060D E008      0380        CPX  #8        ;Reached maximum?
060F 90F5      0390        BCC  LOOP      ;No...do it again
0611 8CC802    0400        STY  COLOR4    ;Yes...put old PCOLR1 into COLOR4
0614 60        0410        RTS            ;Return to BASIC
0615           0420        .END
```

```
1 REM ##############################
2 REM #            DIZZY            #
3 REM #   Original author unknown   #
4 REM #  Modified by Garry Francis  #
5 REM # Published by Atari Computer #
6 REM #    Enthusiasts (N.S.W.)     #
7 REM #        October 1982         #
8 REM ##############################
10 GRAPHICS 10:FOR I=1 TO 8:READ A:POK
E 704+I,50+A:NEXT I
20 DATA 2,4,6,8,10,8,6,4
30 C=1:FOR Y=0 TO 191:COLOR C:PLOT 0,Y
:DRAWTO 79,191-Y
40 C=C+0.416666667:IF C>8 THEN C=1
50 NEXT Y
60 FOR X=79 TO 0 STEP -1:COLOR C:PLOT
X,0:DRAWTO 79-X,191
70 C=C+1:IF C>8 THEN C=1
80 NEXT X:DIM B$(11),C$(1)
90 COLOR 0:XORG=31:YORG=75:B$="ATARI":
GOSUB 150
100 XORG=23:YORG=87:B$="COMPUTER":GOSU
B 150
110 XORG=19:YORG=99:B$="ENTHUSIASTS":G
OSUB 150
120 XORG=28:YORG=112:B$="(N.S.W.)":GOS
UB 150
130 GOSUB 1000
140 X=USR(ADR(A$)):FOR I=1 TO 8:NEXT I
:GOTO 140
150 FOR I=1 TO LEN(B$):RESTORE 200+ASC
(B$(I))
160 READ C$,X,Y:IF C$="P" THEN PLOT XO
RG+X,YORG+Y:GOTO 160
170 IF C$="D" THEN DRAWTO XORG+X,YORG+
Y:GOTO 160
180 IF C$="E" THEN XORG=XORG+X:NEXT I:
RETURN
240 DATA P,1,-3,D,0,-3,D,0,9,D,1,9,E,2
,0
241 DATA P,-1,-3,D,0,-3,D,0,9,D,-1,9,E
,2,0
246 DATA P,0,5,P,0,6,E,2,0
265 DATA P,0;6,D,0,0,D,2,0,D,2,6,P,1,3
,E,4,0
267 DATA P,2,0,D,0,0,D,0,6,D,2,6,E,4,0
269 DATA P,2,0,D,0,0,D,0,6,D,2,6,P,1,3
,E,4,0
272 DATA P,0,0,D,0,6,P,2,0,D,2,6,P,1,3
,E,4,0
273 DATA P,0,0,D,0,6,E,2,0
277 DATA P,0,6,D,0,0,D,4,0,D,4,6,P,2,0
,D,2,6,E,6,0
278 DATA P,0,6,D,0,0,D,2,0,D,2,6,E,4,0
279 DATA P,0,0,D,0,6,D,2,6,D,2,0,D,0,0
,E,4,0
280 DATA P,0,6,D,0,0,D,2,0,D,2,3,D,0,3
,E,4,0
282 DATA P,0,6,D,0,0,D,2,0,D,2,3,P,1,3
,P,1,4,P,2,5,P,2,6,E,4,0
283 DATA P,2,0,D,0,0,D,0,3,D,2,3,D,2,6
,D,0,6,E,4,0
284 DATA P,0,0,D,2,0,P,1,0,D,1,6,E,4,0
285 DATA P,0,0,D,0,6,D,2,6,D,2,0,E,4,0
287 DATA P,0,0,D,0,6,D,4,6,D,4,0,P,2,6
,D,2,0,E,6,0
1000 DIM A$(21):RESTORE 1020
1010 FOR I=1 TO 21:READ A:A$(I)=CHR$(A
):NEXT I:RETURN
1020 DATA 104,162,0,172,193,2,189,194,
2,157,193,2,232,224,8,144,245,140,200,
2,96
```

```
1 REM ##############################
2 REM #           ESCAPE            #
3 REM #   Original author unknown   #
4 REM #  Modified by Garry Francis  #
5 REM # Published by Atari Computer #
6 REM #    Enthusiasts (N.S.W.)     #
7 REM #        October 1982         #
8 REM ##############################
10 GRAPHICS 10
20 FOR I=705 TO 712:READ A:POKE I,224+
A:NEXT I
30 DATA 2,4,6,8,6,4,2,0
40 C=1:FOR I=0 TO 38:COLOR C:X=I:Y=I*2
:PLOT X,Y:DRAWTO 79-X,Y
50 PLOT X,Y+1:DRAWTO 79-X,Y+1:DRAWTO 7
9-X,190-Y:DRAWTO X,190-Y
60 PLOT 79-X,191-Y:DRAWTO X,191-Y:DRAW
TO X,Y
70 C=C+1:IF C>8 THEN C=1
80 NEXT I:GOSUB 1000
90 X=USR(ADR(A$)):FOR I=1 TO 24:NEXT I
:GOTO 90
1000 DIM A$(21):RESTORE 1020
1010 FOR I=1 TO 21:READ A:A$(I)=CHR$(A
):NEXT I:RETURN
1020 DATA 104,162,0,172,193,2,189,194,
2,157,193,2,232,224,8,144,245,140,200,
2,96
```

```
1 REM $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
2 REM $            BRASS            $
3 REM $   Original author unknown   $
4 REM $  Modified by Garry Francis  $
5 REM $ Published by Atari Computer $
6 REM $    Enthusiasts (N.S.W.)     $
7 REM $        October 1982         $
8 REM $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
10 GRAPHICS 9:SETCOLOR 4,15,0
20 FOR Y=0 TO 55 STEP 11:FOR I=0 TO 24
:C=I+3:IF C>15 THEN C=30-C
30 X=Y+I:A=SQR(144-(I-12)*(I-12))/2
40 COLOR 16-C:PLOT X,2*Y+6-A:DRAWTO X,
2*Y+6+A
50 COLOR C:DRAWTO X,Y+130+A:NEXT I:NEX
T Y
60 GOTO 60
```

```
1 REM $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
2 REM $            CONE             $
3 REM $   Original author unknown   $
4 REM $  Modified by Garry Francis  $
5 REM $ Published by Atari Computer $
6 REM $    Enthusiasts (N.S.W.)     $
7 REM $        October 1982         $
8 REM $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
10 GRAPHICS 9:DEG
20 FOR I=-90 TO 269:X=30*SIN(I):Y=160+
30*COS(I):C=ABS(X)/2:IF C<1 THEN C=1
30 IF I>90 THEN COLOR 16-C:GOTO 50
40 COLOR C:PLOT 40,0:DRAWTO 40+X,Y-1
50 PLOT 40,0:DRAWTO 40+X,Y:NEXT I
60 FOR I=0 TO 15:SETCOLOR 4,I,0:FOR J=
1 TO 1000:NEXT J:NEXT I:GOTO 60
```

```
1 REM $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
2 REM $            WHIRL            $
3 REM $   Original author unknown   $
4 REM $  Modified by Garry Francis  $
5 REM $ Published by Atari Computer $
6 REM $    Enthusiasts (N.S.W.)     $
7 REM $        October 1982         $
8 REM $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
10 GRAPHICS 10:FOR I=705 TO 712:READ A
:POKE I,66+A:NEXT I
20 DATA 2,4,6,8,8,6,4,2
30 C=1:FOR Y=0 TO 191:COLOR C:PLOT 0,Y
:DRAWTO 79,191-Y
40 C=C+0.416666667:IF C>8 THEN C=1
50 NEXT Y
60 FOR X=79 TO 0 STEP -1:COLOR C:PLOT
X,0:DRAWTO 79-X,191
70 C=C+1:IF C>8 THEN C=1
80 NEXT X
90 COLOR 0:FOR X=0 TO 20:Y=INT(SQR(420
-X*X))
100 PLOT 39-X,95-Y:DRAWTO 39-X,96+Y:PL
OT 40+X,95-Y:DRAWTO 40+X,96+Y:NEXT X
110 FOR X=0 TO 19:Y=INT(SQR(380-X*X)):
C=8
120 FOR I=95-Y TO 96+Y:COLOR C:PLOT 39
-X,I:PLOT 40+X,I
130 C=C-1:IF C<1 THEN C=8
140 NEXT I:NEXT X:GOSUB 1000
150 X=USR(ADR(A$)):FOR I=1 TO 8:NEXT I
:GOTO 150
1000 DIM A$(21):RESTORE 1020
1010 FOR I=1 TO 21:READ A:A$(I)=CHR$(A
):NEXT I:RETURN
1020 DATA 104,162,0,172,193,2,189,194,
2,157,193,2,232,224,8,144,245,140,200,
2,96
```

```
1 REM $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
2 REM $          GRENHOLE           $
3 REM $   Original author unknown   $
4 REM $  Modified by Garry Francis  $
5 REM $ Published by Atari Computer $
6 REM $    Enthusiasts (N.S.W.)     $
7 REM $        October 1982         $
8 REM $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
10 GRAPHICS 10:POKE 559,0:DEG :DIM S(2
25)
20 FOR I=0 TO 45:A=SIN(I*2):S(I)=A:S(9
0-I)=A:S(90+I)=-A:S(180-I)=-A:S(180+I)
=A:NEXT I
30 FOR I=705 TO 712:READ A:POKE I,208+
A:NEXT I:POKE 559,34
40 DATA 2,4,6,8,6,4,2,0
50 FOR X=0 TO 19:Y=INT(SQR(380-X*X)):C
=8
60 FOR I=95-Y TO 96+Y:COLOR C:PLOT 39-
X,I:PLOT 40+X,I
70 C=C-1:IF C<1 THEN C=8
80 NEXT I:NEXT X:GOSUB 1000
90 TRAP 140:A=20:XOLD=40+A*S(45):YOLD=
96+A*S(0)
100 C=8:FOR I=0 TO 179:X=40+A*S(I+45):
Y=96+A*S(I)
110 COLOR C:PLOT XOLD,YOLD:DRAWTO X,Y:
PLOT XOLD,YOLD+1:DRAWTO X,Y+1
120 A=A+0.025:XOLD=X:YOLD=Y:C=C-1:IF C
<1 THEN C=8
130 Q=USR(ADR(A$)):NEXT I:GOTO 100
140 Q=USR(ADR(A$)):FOR I=1 TO 20:NEXT
I:GOTO 140
1000 DIM A$(21):RESTORE 1020
1010 FOR I=1 TO 21:READ A:A$(I)=CHR$(A
):NEXT I:RETURN
1020 DATA 104,162,0,172,193,2,189,194,
2,157,193,2,232,224,8,144,245,140,200,
2,96
```

```
1 REM ###############################
2 REM #         PSYCHEDELIC         #
3 REM #   Original author unknown   #
4 REM #  Modified by Garry Francis  #
5 REM # Published by Atari Computer #
6 REM #    Enthusiasts (N.S.W.)     #
7 REM #        October 1982         #
8 REM ###############################
10 GRAPHICS 10:POKE 559,0:DEG :DIM S(2
4),C(24)
20 FOR I=0 TO 6:A=SIN(I*15):S(I)=A:S(1
2-I)=A:S(12+I)=-A:S(24-I)=-A
30 A=COS(I*15):C(I)=A:C(12-I)=-A:C(12+
I)=-A:C(24-I)=A:NEXT I
40 FOR I=1 TO 8:POKE 704+I,I*16+10:NEX
T I:POKE 559,34
50 FOR R=1 TO 8:C=R:GOSUB 110:NEXT R
60 FOR R=9 TO 15:C=16-R:GOSUB 110:NEXT
 R
70 FOR R=16 TO 22:C=R-14:GOSUB 110:NEX
T R
80 FOR R=23 TO 29:C=30-R:GOSUB 110:NEX
T R
90 A=PEEK(712)+16:IF A>255 THEN A=26
100 FOR I=705 TO 711:POKE I,PEEK(I+1):
NEXT I:POKE 712,A:GOTO 90
110 COLOR C:I=0:GOSUB 140
120 FOR I=1 TO 24:X2=X1:Y2=Y1:GOSUB 14
0:PLOT X2,Y2:DRAWTO X1,Y1:DRAWTO X2-SG
N(X),Y2:DRAWTO X1,Y1-SGN(Y)
130 NEXT I:RETURN
140 X=(29-R)*S(I):X1=40+X:Y=(60-R)*C(I
):Y1=96+Y:RETURN
```

```
1 REM ###############################
2 REM #            ROLLS            #
3 REM #   Original author unknown   #
4 REM #  Modified by Garry Francis  #
5 REM # Published by Atari Computer #
6 REM #    Enthusiasts (N.S.W.)     #
7 REM #        October 1982         #
8 REM ###############################
10 GRAPHICS 10:DEG
20 FOR I=705 TO 712:POKE I,134:NEXT I
30 FOR I=180 TO 534 STEP 6:X=9*COS(I):
Y=70+8*SIN(I):C=INT((I-180)/45)+1
40 COLOR 0:PLOT X+9,Y:COLOR C:DRAWTO X
+9,Y+50
50 COLOR 0:PLOT X+29,Y:COLOR 9-C:DRAWT
O X+29,Y+50
60 COLOR 0:PLOT X+49,Y:COLOR C:DRAWTO
X+49,Y+50
70 COLOR 0:PLOT X+69,Y:COLOR 9-C:DRAWT
O X+69,Y+50
80 NEXT I:POKE 705,142
90 A=PEEK(705):FOR I=705 TO 711:POKE I
,PEEK(I+1):NEXT I:POKE 712,A:GOTO 90
```

```
1 REM ###############################
2 REM #           STRIPES           #
3 REM #   Original author unknown   #
4 REM #  Modified by Garry Francis  #
5 REM # Published by Atari Computer #
6 REM #    Enthusiasts (N.S.W.)     #
7 REM #        October 1982         #
8 REM ###############################
10 GRAPHICS 10:FOR I=705 TO 712:READ A
:POKE I,160+A:NEXT I
20 DATA 2,4,6,8,6,4,2,0
30 C=1:FOR Y=0 TO 62 STEP 2:FOR I=0 TO
 8:FOR J=4 TO 58 STEP 18:X=I+J:X1=X+9
40 COLOR C:PLOT X,Y:PLOT X,Y+1:PLOT X1
,Y+64:PLOT X1,Y+65:PLOT X,Y+128:PLOT X
,Y+129
50 COLOR 9-C:PLOT X1,Y:PLOT X1,Y+1:PLO
T X,Y+64:PLOT X,Y+65:PLOT X1,Y+128:PLO
T X1,Y+129:NEXT J
60 C=C+1:IF C>8 THEN C=1
70 NEXT I:NEXT Y:GOSUB 1000
80 X=USR(ADR(A$)):FOR I=1 TO 18:NEXT I
:GOTO 80
1000 DIM A$(21):RESTORE 1020
1010 FOR I=1 TO 21:READ A:A$(I)=CHR$(A
):NEXT I:RETURN
1020 DATA 104,162,0,172,193,2,189,194,
2,157,193,2,232,224,8,144,245,140,200,
2,96
```

```
1 REM ###############################
2 REM #          HYPNOSIS           #
3 REM #   Original author unknown   #
4 REM #  Modified by Garry Francis  #
5 REM # Published by Atari Computer #
6 REM #    Enthusiasts (N.S.W.)     #
7 REM #        October 1982         #
8 REM ###############################
10 GRAPHICS 10
20 FOR I=1 TO 8:POKE 704+I,32*I-10:NEX
T I
30 C=1:FOR Y=0 TO 191:COLOR C:PLOT 0,Y
:DRAWTO 79,191-Y
40 C=C+0.416666667:IF C>8 THEN C=1
50 NEXT Y
60 FOR X=79 TO 0 STEP -1:COLOR C:PLOT
X,0:DRAWTO 79-X,191
70 C=C+1:IF C>8 THEN C=1
80 NEXT X:GOSUB 1000
90 X=USR(ADR(A$)):FOR I=1 TO 5:NEXT I:
GOTO 90
1000 DIM A$(21):RESTORE 1020
1010 FOR I=1 TO 21:READ A:A$(I)=CHR$(A
):NEXT I:RETURN
1020 DATA 104,162,0,172,193,2,189,194,
2,157,193,2,232,224,8,144,245,140,200,
2,96
```

```
1 REM ################################
2 REM #          BEACH BALL          #
3 REM #   Original author unknown    #
4 REM #  Modified by Garry Francis   #
5 REM # Published by Atari Computer  #
6 REM #    Enthusiasts (N.S.W.)      #
7 REM #         October 1982         #
8 REM ################################
10 GRAPHICS 10:POKE 559,0:DEG :DIM S(2
4),C(24)
20 FOR I=0 TO 6:A=SIN(I*15)/2:S(I)=A:S
(12-I)=A:S(12+I)=-A:S(24-I)=-A
30 A=COS(I*15)*80:C(I)=A:C(12-I)=-A:C(
12+I)=-A:C(24-I)=A:NEXT I
40 FOR I=705 TO 712:READ A:POKE I,A:NE
XT I:POKE 559,34
50 DATA 14,14,52,52,118,118,184,184
60 A1=0:FOR C=1 TO 8:A2=40-40*COS(C*11
.25)
70 FOR R=A1 TO A2:COLOR C:PLOT 40,176
80 FOR I=1 TO 24:X=40+(40-R)*S(I):Y=96
+C(I)
90 IF I=13 THEN COLOR 9-C
100 DRAWTO X,Y:NEXT I:NEXT R:A1=A2:NEX
T C
110 A=PEEK(705):FOR I=705 TO 711:POKE
I,PEEK(I+1):NEXT I:POKE 712,A:GOTO 110
```

```
1 REM ################################
2 REM #          WATER MELON         #
3 REM #   Original author unknown    #
4 REM #  Modified by Garry Francis   #
5 REM # Published by Atari Computer  #
6 REM #    Enthusiasts (N.S.W.)      #
7 REM #         October 1982         #
8 REM ################################
10 GRAPHICS 10:POKE 559,0:DEG :DIM S(3
6)
20 FOR I=0 TO 9:A=SIN(10*I):S(I)=A:S(1
8-I)=A:S(18+I)=-A:S(36-I)=-A:NEXT I
30 FOR I=705 TO 712:READ A:POKE I,A:NE
XT I:POKE 559,34
40 DATA 198,200,202,204,204,202,200,56
50 X1=5:Y1=96:F=0:GOSUB 100
60 X1=74:F=1:GOSUB 100
70 GOTO 70
100 C=1:A=40
110 COLOR C:PLOT X1,Y1:FOR I=18 TO 27:
X=25*S(I+9)+30:IF F THEN X=79-X
120 Y=A*S(I):DRAWTO X,96+Y:NEXT I
130 C=C+1:IF C>7 THEN C=1
140 A=A-1:IF A>-40 THEN 110
150 COLOR 8:FOR I=0 TO 18:X=4*S(I+9)+3
0:IF F THEN X=79-X
160 Y=40*S(I):PLOT X,96+Y:DRAWTO X,96-
Y:NEXT I
170 COLOR 0:FOR I=1 TO 25:X=RND(0)*6+2
7:IF F THEN X=79-X
180 Y=RND(0)*60+66:PLOT X,Y:NEXT I:RET
URN
```

```
1 REM ################################
2 REM #             SAS              #
3 REM #   Original author unknown    #
4 REM #  Modified by Garry Francis   #
5 REM # Published by Atari Computer  #
6 REM #    Enthusiasts (N.S.W.)      #
7 REM #         October 1982         #
8 REM ################################
10 GRAPHICS 10:POKE 559,0:DEG :DIM S(4
50)
20 FOR I=0 TO 90:A=SIN(I):S(I)=A:S(180
-I)=A:S(180+I)=-A:S(360-I)=-A:S(360+I)
=A:NEXT I
30 FOR I=705 TO 712:READ A:POKE I,114+
A:NEXT I
40 DATA 2,4,6,8,8,6,4,2
50 GOSUB 1000:C=1:POKE 559,34
60 FOR I=0 TO 359:X=40+30*S(90+I):Y=96
+80*S(I):COLOR INT(C)
70 PLOT 65,97:DRAWTO X,Y+1:PLOT 65,96:
DRAWTO X,Y:PLOT 65,95:DRAWTO X,Y-1
80 C=C+0.5:IF C>8.5 THEN C=1
90 X=USR(ADR(A$)):NEXT I
100 X=USR(ADR(A$)):FOR I=1 TO 14:NEXT
I:GOTO 100
1000 DIM A$(21):RESTORE 1020
1010 FOR I=1 TO 21:READ A:A$(I)=CHR$(A
):NEXT I:RETURN
1020 DATA 104,162,0,172,193,2,189,194,
2,157,193,2,232,224,8,144,245,140,200,
2,96
```

```
1 REM ################################
2 REM #            POLES             #
3 REM #   Original author unknown    #
4 REM #  Modified by Garry Francis   #
5 REM # Published by Atari Computer  #
6 REM #    Enthusiasts (N.S.W.)      #
7 REM #         October 1982         #
8 REM ################################
10 GRAPHICS 10:FOR I=705 TO 712:READ A
:POKE I,50+A:NEXT I
20 DATA 2,4,6,8,6,4,2,0
30 C=1:FOR Y=0 TO 58 STEP 2:FOR I=0 TO
 8:COLOR C
40 FOR J=5 TO 65 STEP 12:X=I+J:PLOT X,
Y:PLOT X,Y+1:PLOT X,Y+64:PLOT X,Y+65:P
LOT X,Y+128:PLOT X,Y+129:NEXT J
50 C=C+1:IF C>8 THEN C=1
60 NEXT I:NEXT Y:GOSUB 1000
70 X=USR(ADR(A$)):FOR I=1 TO 18:NEXT I
:GOTO 70
1000 DIM A$(21):RESTORE 1020
1010 FOR I=1 TO 21:READ A:A$(I)=CHR$(A
):NEXT I:RETURN
1020 DATA 104,162,0,172,193,2,189,194,
2,157,193,2,232,224,8,144,245,140,200,
2,96
```

ATARI (tm) GRAPHICS AND SOUND MADE EASY!

SANTA CRUZ EDUCATIONAL SOFTWARE HAS WRITTEN A SERIES OF AFFORDABLE PROGRAMS THAT DEMONSTRATES MANY OF THE SPECIAL "TRICKS" THAT ONLY THE ATARI 400/800 COMPUTER IS CAPABLE OF DOING. WE OFFER EVERYTHING FROM A PROGRAM THAT DOES FANTASTIC HIGH RESOLUTION GRAPHICS TO ONE THAT ACTUALLY DIALS YOUR PHONE.....

****

MASTER MEMORY MAP(tm) – This is really the key to using the ATARI'S capabilities We start out by explaining how to PEEK and POKE values into memory so that even new programmers can use this Then, we give you over 15 pages of the memory locations that are the most useful. The information is condensed from both the ATARI'S Operating System Manual and various articles and programs. It is, of course, useful even for experienced programmers as a reference. Also, we highly suggest that dealers offer this Memory Map to customers who request to be told how to use the power of the machine. We guarantee it will answer many of the questions you have about the machine.

$6.95

TRICKY TUTORIALS(tm)

#1: DISPLAY LISTS – This program teaches you how to alter the program in the ATARI that controls the format of the screen. For example: when you say graphics 8 the machine responds with a large graphics 8 area at the top of the screen and a small text area at the bottom. Now, you will be able to mix the various modes on the screen at the same time. Just think how nice your programs could look with a mix of large and small text, and both high and low resolution graphics, this program has many examples plus does all of the difficult caluculations!

#2: HORIZONTAL/VERITCAL SCROLLING – The information you put on the screen, either graphics or text, can be moved up, down or sideways. This can make some nice effects. You could move only the text on the bottom half of the screen or perhaps create a map and then move smoothly over it by using the joystick.

#3: PAGE FLIPPING – Normally you have to redraw the screen every time you change the picture or text. Now you can learn how to have the computer draw the next page you want to see while you are still looking at the previous page, then flip to it instantly. You won't see it being drawn, so a complicated picture can seem to just appear. Depending on your memory size and how complicated the picture, you could flip between many pages, thus allowing animation or other special effects with your text.

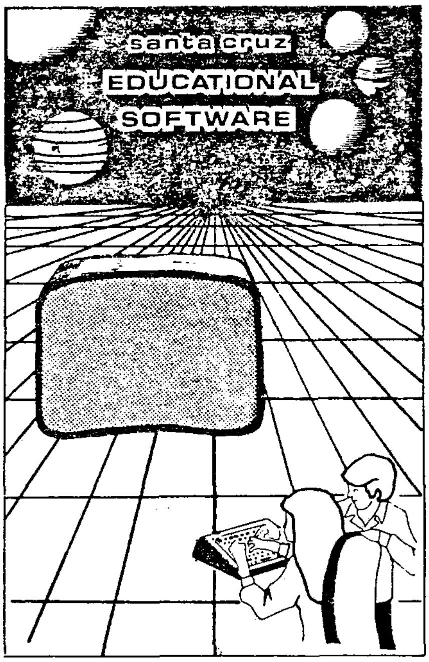
#4: BASICS OF ANIMATION – Shows you how to animate simple shapes using the PRINT and PLOT commands, and also has nice little PLAYER/MISSILE Graphics demo to learn. This would be an excellent way to start making your programs come alive on the screen. Recommended for new owners.

#5 PLAYER MISSILE GRAPHICS – This complex subject is demonstrated by starting with simple examples, and building up to a complete game and also an animated business chart on multiple pages! As always, the computer does most of the calculations. Requires 32K disk or tape and costs **$29.95**

#6: SOUND – From explaining how to create single notes, to demonstrating complex four channel sound effects, this newest tutorial is great. Even those experienced with ATARI's sound capabilities will find the menu of sound effects a needed reference that can be used whenever you are in the need of a special sound for your programs. Everyone will learn something new! Written by Jerry White.

Tricky Tutorials (except #5) require 16K memory for cassette orders and 24K for disk. The price is $19.95each. You may order 1,2,3, & 4 for $64.95.All six in a colorful binder cost $99.95.

THE GRAPHICS MACHINE!! – Turn your computer into an incredible graphics tool with advanced commands like circle, box, fill, polygon, line, help, etc. 3 colors in graphics 8 with instant text!!! Create colorful business charts or beautiful drawings and then save or retrieve them from disk in 5 SECONDS. YES, it's that fast. Needs all 48K, disk, and costs **$19.95**



MINI-WORD PROCESSOR – This is for those of you who have a printer, but don't want to spend $100 or more for a fancy word processor. It is suitable for simple editing of text, accepts most control characters for your printer, and text is stored on disk for easy retrieval. Holds 2½ typed pages at a time. Requires 32K, disk or tape. **$19.95**

BOB'S BUSINESS – 14 small business type programs for home or office, all chosen from a nice menu. Supports printed output. 169 sectors of output require 16K tape, or 32K disk. **$14.95**

KID'S #1 – Includes the following 1) TREASURE – search for the lost treasure while trying to keep from falling into the sea. Nice graphics if you find it!, 2)DIALOGUE – talk back to the computer about four subjects: 3) MATH QUIZ – Nice musical and graphical rewards for good scores Parents input the level of difficulty

KID'S #2 – A spelling quiz, a "scrabble" type game, and a version of Touch with the computer giving all the directions! Both Kid's programs require 16K tape or 24K disk and cost **$14.95** each.

MINI-DATABASE/DIALER – This unique new program stores and edits up to 8 lines of information such as name, address, and phone numbers, or messages, inventories or anything you want. It has the usual sort, search, and print options, but it also has an unusual feature: If your files include phone numbers and you have a touch-tone phone, the program will DIAL THE PHONE NUMBERS FOR YOU! This is perfect for those who make a lot of calls like salesmen, teens, or those trying to get through to busy numbers (acts as an auto-redialer). It is also a lot of fun to use. Requires 16K cassette or 24K disk and costs **$24.95**

FONETONE – For those who only want to store name and phone numbers and have the dialer feature as above, we offer this reduced version Same memory requirements, but only costs **$14.95**. Don't forget you must have a touch-tone phone.

PLAYER PIANO – Turns your keyboard into a mini-piano and more. Multiple menu options provide the ability to create your own songs, save or load data files using cassette or diskette, fix or change any of up to 400 notes in memory, and play all or part of a song The screen displays the keyboard and indicates each key as it is played from a data file or the notes you type. You don't have to be a musician to enjoy this educational and entertaining program. Requires 24K cassette or 32K disk. **$14.95**

All

available

from

COMPUTERWAY

PTY LIMITED

All Correspondence to:
Box 268
GPO Sydney
N.S.W. 2001

Lower Ground Floor
Myer Sydney Store
George and Market Streets
Sydney
Tel: 238 9984

Educational Software • Tricky Tutorials • Santa Cruz Educational Software • Tricky Tutorials • Santa Cruz Educational Software • Tricky Tutorials • Santa Cruz Educational Software • Tricky Tutorials • Santa Cruz Educational Software

• Santa Cruz Educational Software • Tricky Tutorials • Santa Cruz Educational Software • Tricky Tutorials • Santa